AD-A088 084 NAVAL RESEARCH LAB WASHINGTON DC F/G 9/2
A HIGHLY PORTABLE DATA ACQUISITION SYSTEM FOR TOTAL MAGNETICS F--ETC(U)
AUG 80 D STEIGER
UNCLASSIFIED NRL-MR-4300 NL

1 OF 1
AD
A088084

END
DATE
FILMED
9-80
DTIC

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

9

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
NRL Memorandum Report 4300

2. GOVT ACCESSION NO.
AD-A088084

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
A HIGHLY PORTABLE DATA ACQUISITION SYSTEM FOR TOTAL MAGNETICS FIELD MEASUREMENTS.

6

5. TYPE OF REPORT & PERIOD COVERED
Interim report on a continuing NRL problem

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)

10

Daniel Steiger

12 27

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Naval Research Laboratory
Washington, D.C. 20375

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
68011-6013

11. CONTROLLING OFFICE NAME AND ADDRESS

14 NRL-MR-4300

11

12. REPORT DATE
8 Aug[redacted]80

13. NUMBER OF PAGES
26

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

15. SECURITY CLASS. (of this report)
UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

DTIC
S ELECTE D
AUG 1 9 1980
B

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Intelligent terminals
Real-Time processing
Microcomputer
Magnetics data acquisition
Portable data acquisition system

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

At the Naval Research Laborat y highly portable intelligent terminals are being utilized aboard oceanographic platforms for the acquisition and storage of total magnetic field measurements. Described in the paper is the design and implementation of the data acquisition system which includes a detailed description of the hardware and software utilized.

CONTENTS

A HIGHLY PORTABLE DATA ACQUISITION SYSTEM FOR TOTAL MAGNETICS FIELD MEASUREMENTS

I. INTRODUCTION

The Acoustic Media Characterization Branch of the Acoustics Division of NRL has since 1971 conducted magnetic field studies for the Navy in many areas of the world. Much of this work has been accomplished using aircraft to conduct the magnetic studies. The resultant data has been compiled into magnetic anomaly charts of the oceans of the world. The information is utilized by DOD for interpretation of the tectonic history of the area. These interpretations utilize state of the art theories of sea floor spreading and plate tectonics.

Many of these magnetic studies are performed on aircraft and ships of opportunity or "piggybacking" other experiments. During some of these data gathering experiments it is inappropriate or impractical to place a mini-computer system aboard the aircraft due to size, weight or the resources required to install, maintain and operate a computer system.

NRL has developed a highly portable, compact, lightweight and easy to operate data acquisition system for acquiring magnetics field data and navigational position information. The portable magnetics data acquisition system has been used successfully yielding results comparable to the mini-computer system.

II. PORTABLE DATA ACQUISITION SYSTEM

The portable data acquisition system is based around a Hewlett-Packard HP264X intelligent terminal. Figure 1 is a photograph of the terminal. The feature of the terminal that makes it intelligent is the programmable microprocessor internal to the terminal. Also, all of the necessary components of a computer such as internal storage, external storage and interface capability to peripherals provide the capability of making the terminal a data acquisition system.

The HP264X was selected as the data acquisition system because it had all of the salient features required such as, programmable processor, internal memory, external cassette storage, parallel and serial interfacing capability, portability and software development tools. In addition, several of these units are available at NRL and being used as computer terminals. In fact, the HP264X can be considered a well packag-

Manuscript submitted June 11, 1980.

ed, self-contained microcomputer.

A functional diagram of the portable terminal magnetics field data acquisition system is shown in Figure 2. The system consists of a Geometrics Model G801/803A magnetometer, a CHRONO-LOG Series 70000 Time Code Generator and a NRL designed Litton Interface that obtains aircraft position information from the aircraft Litton Model 72 Inertial Navigator. The components of the system are described below.

1. Terminal Electronic Circuit Boards

An interior view of the terminal is shown in Figure 3. There are fifteen circuit boards that can be inserted directly into the terminal. Ten of these boards are required for control of the terminal, with the microprocessor residing on one of these boards. To use the HP264X as a data acquisition device two high density HP13297A-003 32K Byte RAM (Random Access Memory) Boards are required. The strapping configuration of these boards are given in Table 1. This memory is used for display, programs, temporary data storage and assembling and debugging programs. The HP264X in this configuration leaves five empty slots for interfacing to external sensors.

2. I/O Terminal Interfaces

The HP13255 Terminal Duplex Register Board described in Reference 1 was selected for interfacing with the magnetometer, digital clock and Litton Inertial Navigator Interface. All of these devices provide Binary Coded Decimal (BCD) outputs at TTL logic levels. The Duplex Register Board contains 8 data receiving lines and 8 status lines. It was recognized that the 8 status lines could be used for data input as well as the 8 data lines resulting in 16 data lines for input. The polarity of the status lines on the interface is reversed from the data lines except for bits zero and one. By using the status lines the input capacity was increased from five 8 bit words to five 16 bit words, thereby doubling the data acquisition capacity of the terminal. Also, by using the status lines the 16 bit four character BCD output of the magnetometer and Litton Navigator Interface was fully compatible. The problem of polarity was handled with software by masking the two status bits of opposite polarity complementing the remainder and adding the two bits to the remainder to reform the byte.

3. External Storage

The HP264X Terminal has two cassette drive units mounted below the display. Each cassette is capable of storing 110K Bytes of information. The information stored on these cassettes are source, object, assembler and debugger programs and the data acquired from the magnetics

data acquisition system. The cassettes can be operated using functional keys from the keyboard or under program control. Both ASCII and binary tapes can be read and written by the terminal.

4. Litton Interface

The interface between the Litton Inertial Navigator and the data acquisition system was specially designed and built at NRL. The Litton Navigator sequentially outputs inertial navigation information on a continuous basis. The function of the interface is to service a request for data from the data acquisition system. Upon request from the controller the interface obtains and stores latitude, longitude and heading information. After acquiring this information the interface interrupts the terminal data acquisition system. Upon receiving the interrupt the terminal goes through a "handshake" sequence with the interface three times to acquire the latitude, longitude and heading which is sequentially multiplexed to the output lines of the interface.

The interface is connected to the terminal using two duplex boards. Two boards are required since eight BCD characters form a position word. The terminal I/O board addresses of the four most significant characters and the four least significant characters are given in Table 2. The addresses are accomplished by configuring jumpers at appropriate locations on the duplex board.

5. Magnetometer

The magnetometer measures the magnetic field intensity at either preset or continuous intervals. Five BCD TTL compatible digital characters are output through a connector on the back of the unit to the terminal and also output to the display of the magnetometer. Since each terminal interface is capable of accepting four BCD characters two interfaces are required. The most significant character from the magnetometer is interfaced to one eight bit duplex board while the four remaining characters are interfaced to a second duplex board in the terminal. The I/O strapping configurations are given in Table 2. The strapping consists of assigning the board an address that can be read by the program. The addresses of each respective board is given in Table 2.

6. Digital Clock

The function of the clock is to provide digital time in order to be able to correlate and interpolate data when future processing is performed. The clock provides in BCD format at TTL signal levels day of year, hours, minutes and seconds. In order to conserve input capacity, minutes and seconds consisting of four BCD characters were interfaced to the terminal using one duplex board. Day and hour are hand recorded on the cassette cartridge and time information is reconstructed when

further processing is performed. The address of the terminal interface board is given in Table 2. Also, it should be noted that there are several manufacturers of digital clocks which can be used and have been used since they function similarly to the CHRONO-LOG.

III. SOFTWARE DESCRIPTION

Programs for the intelligent terminal can be developed by preparing the source program and using the assembler available on the terminal or by using an HP1000 mini-computer system to prepare the program and provide a cross assembly for loading into the terminal. Since the debugging of the program can only be performed on the terminal the program for the terminal magnetics system was developed on the terminal.

The terminal uses a Intel 8080 compatible microprocessor. The microprocessor differences are in the way I/O is managed. Therefore, the program with the exception of I/O is Intel 8080 compatible. The terminal has many software subroutines stored in Read Only Memory (ROM) that can be used by the program by addressing the starting location of the subroutines. These subroutines, since they are stored in ROMs can not be altered. The routine PUTIO for performing I/O to the terminal display and cartridge tape units was used. This routine will write ASCII records to the display and either tape drive depending upon the device specified. The terminal magnetics program has been programmed to use only the right tape drive to store data.

The terminal has a 10 millisecond internal clock. The clock is used to schedule the magnetics program by storing the number of 10 millisecond intervals required in a location called TIMER which the terminal executive system decrements. Upon decrementing the location to zero the executive system software transfers control to a predetermined location. The starting address of the user program is stored at this location which in turn permits the scheduling of subroutines. The magnetics data acquisition program was scheduled to execute every three seconds. This required the storing of 100 in the location TIMER which equates to one second and executing the timer program three times. This was required since the microprocessor is organized around an 8 bit word which has 127 as its largest positive number.

The terminal data acquisition program is entered by transferring control from the terminal executive program to the program CHTIMO. The function of CHTIMO is to schedule the data acquisition program to run at three second intervals. This is accomplished by checking for the TIMER location to go to zero and the number of repetitive seconds to go negative. When the repetitive seconds have expired software control transfers to the main program CONTRL, otherwise a return to the terminal executive program is executed.

The program CONTRL is used to call four major subroutines, namely, INIT2, INPUT, PROCES and OUTPUT. These four programs are discussed below.

1. Subroutine INIT2

The program INIT2 stores 100 in the location TIMER which allows the terminal executive system to decrement the location TIMER 100 times, which takes one second before going to zero. Also, the program sets the repeat factor of this program at two in order to obtain three second intervals between the magnetics program execution. The program INIT2 is called every time the program CHTIMO calls the program CNTRL.

2. Subroutine INPUT

The function of subroutine INPUT is to obtain the data from the external sensors and devices. It accomplishes this task by requesting data from the devices using a memory mapped I/O scheme. All of the five interface boards in the terminal have a unique address determined by the strapping configuration on the board which are given in Table 2. Under program control a request is made of the sensor, or sensor interface to send data. The data is buffered into the terminal interface I/O board. By addressing the terminal interface board with its unique address the data can be handled by the microprocessor under program control.

In the case of the Litton navigation information the process is repeated three times since the data is multiplexed out using the same two interface boards in order to obtain latitude, longitude and heading of the aircraft.

3. Subroutine PROCES

The program PROCES is used to manipulate the data into a format suitable for display and storage on cassette tapes. The first step in the process of preparing the data for output is to convert the BCD characters to an ASCII format. The status byte consisting of bits zero and one being of opposite polarity to the remaining word must be complemented and the word reformed. Following all the status words and data words being in the same BCD format, the data is manipulated by an algorithm that replaces the BCD character with its equivalent ASCII character.

In the case of latitude and longitude the first bit of the status byte is masked and tested for 0 or 1, which determines North or South for latitude and East and West for longitude.

4. Subroutine OUTPUT

In order to output the data to the display for monitoring and

the cartridge tape for storage a terminal executive system routine called PUTIO is utilized. The program moves the data in ASCII format to a terminal system output buffer and PUTIO is called. PUTIO places the data on the display and the right cartridge tape.

IV. PROGRAM DEVELOPMENT

The source program is written in a compatible INTEL 8080 language with the only exception being the I/O operations. These I/O operations are accomplished using programs stored in a terminal ROM.

1. Preparing the Program

For assembling and loading, the source and binary programs must reside on cartridge tape. The source program can be placed on the tape by entering the source code into the terminal display memory through the terminal keyboard. Once in the display memory the source code is transferred to tape using the terminal function keys which provide the capability to transfer data between the terminal and other devices. An alternate method of obtaining the source code on tape is by keying the program into a file using the HP1000 mini-computer system. The file can then be edited and "dumped" to cartridge tape in ASCII format.

2. Assembling the Program

The HP13290B Debugger/Assembler is a commercially available product from Hewlett-Packard, and it resides on cartridge tape. By placing the tape in the left drive of the terminal it is loaded using the function keys on the terminal. Once having loaded the assembler the source program which resides on tape is placed on the left drive and a blank tape to receive the assembled code in the right drive. After having successfully completed the assembly the right tape with the assembled code is then placed in the left tape drive and under keyboard command is loaded into the terminal. At this point the program is ready for execution. Operating instructions for the magnetics data acquisition system are given in Appendix II. An alternate manner of assembling the program is to use the cross-compiler available on the HP1000 mini-computer system. The assembled program is stored on tape in the same format as the assembly on the terminal. Refer to Reference 2 for specific instruction on using the HP13290B Debugger/Assembler.

V. RESULTS

The major benefits of the Magnetic Field Terminal Data Acquisition System is its compactness (all data acquisition components are integrated into the terminal), weight of 45 pounds and reliability. This can be compared with the mini-computer system which resides in a 56 inch equip-

ment rack which weighs approximately 600 pounds.

Experiments collecting magnetic field data have shown the accuracy of the terminal data acquisition system is identical to the accuracy of the mini-computer system. However, when the mini-computer system is utilized, the magnetics data can be processed to completion, whereas, the data stored by the terminal data acquisition system must be further processed by the mini-computer at some future time. Additionally, the storage capacity of the mini-computer system is far greater than the terminal resulting in the mini-computer being operated for much longer periods of time before the data must be stored in another manner such as nine track 800bpi magnetic tape.

The terminal Magnetics Field Data Acquisition System has been proven to be a viable alternative to the mini-computer when the mini-computer system is inappropriate. The terminal systems have been utilized to acquire and process data for Airborne Expendable Bathythermograph (AXBT) experiments and recording the environment during acoustic studies as well as magnetic field experiments.

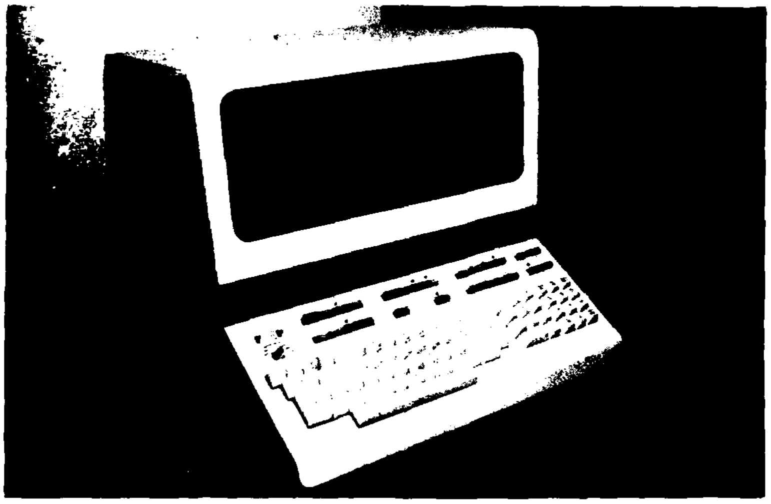

Fig. 1 — Photograph of HP264X Intelligent Terminal

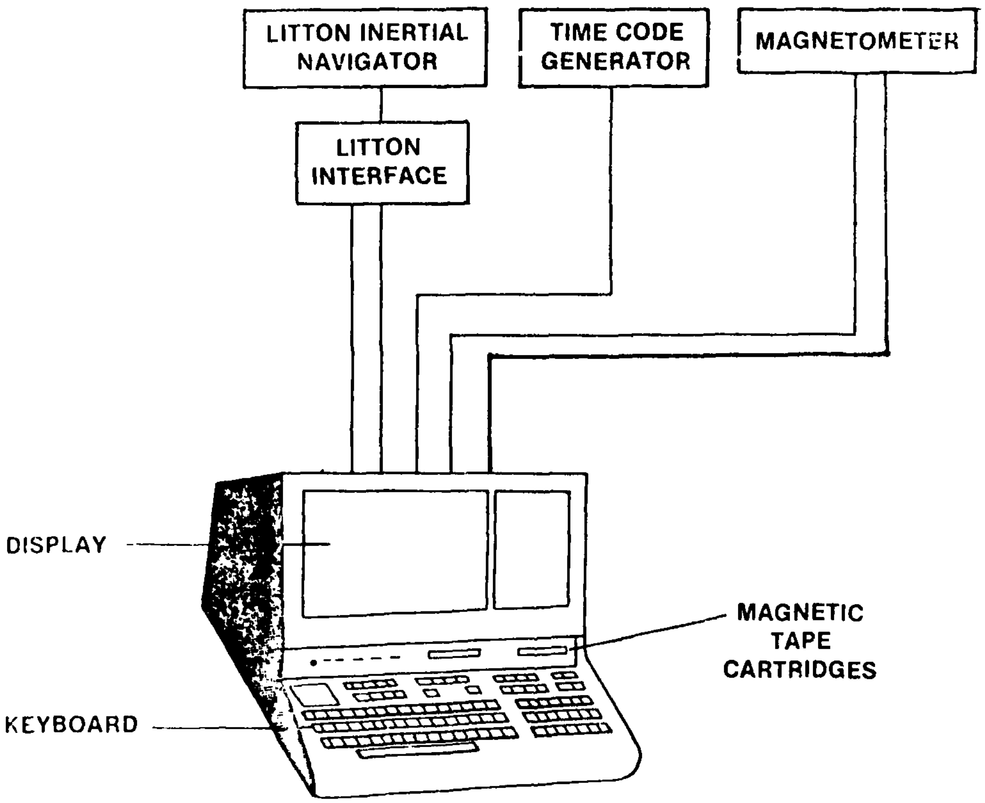

Fig. 2 — Functional Block Diagram of Terminal Magnetics Data Acquisition System

Fig. 3 — Interior View of HP264X Intelligent Terminal

TABLE 1

SWITCH POSITIONS FOR HP13297A-003 32K
BYTE RAM

SWITCH	BOARD 1	BOARD 2
INH	OPEN	OPEN
32K	OPEN	CLOSED
16K	OPEN	OPEN
8K	OPEN	OPEN
4K	OPEN	OPEN
INH	OPEN	OPEN
32K	OPEN	CLOSED
16K	CLOSED	CLOSED
8K	OPEN	OPEN
4K	OPEN	OPEN
R.M	OPEN	OPEN
RAM	OPEN	OPEN
R.M	OPEN	OPEN
RAM	OPEN	OPEN
M1	CLOSED	CLOSED
.M2	OPEN	OPEN
.M3	OPEN	OPEN
FST	OPEN	OPEN
RPT	OPEN	OPEN
WPT	OPEN	OPEN

TABLE 2

JUMPER CONNECTIONS FOR HP13255
TERMINAL DUPLEX BOARDS

DEVICE	A	B	C	D	E	F	G	H	J	K	L	M	N	P	Q	R	ADDRESS
MOST SIGNIFICANT CHARACTERS LITTON	OUT	IN	OUT	OUT	IN	IN	OUT	IN	IN	OUT	IN	IN	OUT	OUT	IN	OUT	102XXX
CLOCK	OUT	IN	OUT	OUT	IN	OUT	OUT	IN	IN	OUT	IN	IN	OUT	OUT	IN	OUT	103XXX
LEAST SIGNIFICANT CHARACTERS MAGNETOMETER	OUT	IN	OUT	OUT	IN	IN	IN	OUT	IN	OUT	IN	IN	OUT	OUT	IN	OUT	104XXX
LEAST SIGNIFICANT CHARACTERS LITTON	OUT	IN	OUT	OUT	IN	OUT	IN	OUT	IN	OUT	IN	IN	OUT	OUT	IN	OUT	105XXX
MOST SIGNIFICANT CHARACTERS MAGNETOMETER	OUT	IN	OUT	OUT	IN	IN	OUT	OUT	IN	OUT	IN	IN	OUT	OUT	IN	OUT	106XXX

APPENDIX I

SOURCE LISTING TERMINAL MAGNETICS PROGRAM

```
.PROGRAM TO ACQUIRE MAGNETICS DATA,PREPARED FOR ASCENSION 12/04/79
ALTIO   EQU 20B          ;DEFINES PROGRAM AS ALTERNATE I/O DRIVER
TIMER   EQU 176147Q      ;TIME OUT COUNTER
PUTIO   EQU 4199H        ;SUBROUTINE TO OUTPUT DATA IN ASCII FMT
OUTDEV  EQU 0FF4DH       ;SPECIFIES OUTPUT DEVICE
GTIOB0  EQU 3D1BH        ;SYSTEM SUBROUTINE TO GET AN I/O BUFFER
GETPTR  EQU 3D46H        ;SYSTEM BUFFER ADDRESS
XFRLIM  EQU 0FF47H       ;SPECIES THE NUMBER OF CHAR FOR OUTPUT
INDVM   EQU  106001Q     ;ADDRESS TO INPUT DATA OF MSCHARS MAGGIE
INSTAD  EQU  106000Q     ;ADDRESS TO INPUT STATUS OF MSCHARS MAGGIE
INITD   EQU  106007Q     ;SETS IN FF ON MSCHARS MAGGIE I/O BOARD
RSTDVM  EQU  106005Q     ;RESETS IN FF ON MSCHARS MAGGIE I/O BOARD
STOUTD  EQU  106006Q     ;SETS OUT FF ON MSCHARS MAGGIE I/O BOARD
RSOUTD  EQU  106004Q     ;RESETS OUT FF ON MSCHARS MAGGIE I/O BOARD
INCLK   EQU   103001Q    ;ADDRESS TO INPUT DATA(SECONDS) OF CLOCK
INSTAC  EQU   103000Q    ;ADDRESS TO INPUT STATUS(MIN) OF CLOCK
INITC   EQU   103007Q    ;SETS IN FF ON CLOCK I/O BOARD
RSTCLK  EQU   103005Q    ;RESETS IN FF ON CLOCK I/O BOARD
STOUTC  EQU   103006Q    ;SETS OUT FF ON CLOCK I/O BOARD
RSOUTC  EQU   103004Q    ;RESETS OUT FF ON CLOCK I/O BOARD
INMAG   EQU   104001Q    ;ADDRESS TO INPUT DATA OF LSCHARS MAGGIE
INSTAM  EQU   104000Q    ;ADDRESS TO INPUT STATUS OF LSCHARS MAGGIE
INITM   EQU   104007Q    ;SETS IN FF ON LSCHARS MAGGIE I/O BOARD
RSTMAG  EQU   104005Q    ;RESETS IN FF ON LSCHARS MAGGIE I/O BOARD
STOUTM  EQU   104006Q    ;SETS OUT FF ON LSCHARS MAGGIE I/O BOARD
RSOUTM  EQU   104004Q    ;RESETS OUT FF ON LSCHARS MAGGIE I/O BOARD
INLIT   EQU   105001Q    ;ADDRESS TO INPUT DATA LSCHARS LITTON I/O BD
INSTAL  EQU   105000Q    ;ADDRESS TO INPUT STATUS LSCHARS LITTON I/O BD
INITL   EQU   105007Q    ;SETS IN FF ON LSCHARS LITTON I/O BOARD
RSTLIT  EQU   105005Q    ;RESETS IN FF ON LSCHARS LITTON I/O BOARD
STOUTL  EQU   105006Q    ;SETS OUT FF ON LSCHARS LITTON I/O BOARD
RSOUTL  EQU   105004Q    ;RESETS OUT FF ON LSCHARS LITTON I/O BOARD
CHKFF   EQU   105003Q    ;ADDRESS TO READ FLAG ON LSCHARS LITTON I/O BD
INLIT1  EQU   102001Q    ;ADDRESS TO INPUT DATA ON MSCHARS LITTON I/O BD
INSTL1  EQU   102000Q    ;ADDRESS TO INPUT STATUS ON MSCHARS LITTON I/O BD
INITL1  EQU   102007Q    ;SETS IN FF ON MSCHARS LITTON I/O BOARD
RSTLT1  EQU   102005Q    ;RESETS IN FF ON MSCHARS LITTON I/O BOARD
STOTL1  EQU   102006Q    ;SETS OUT FF ON MSCHARS LITTON I/O BOARD
RSOTL1  EQU   102004Q    ;RESETS OUT FF ON MSCHARS LITTON I/O BD
INFF    EQU   80H        ;MASK TO CHECK RESET STATUS ON LSCHARS LITTON BD
MASK1   EQU   17Q        ;MASKS FOUR LSBITS,USED IN BCD TO ASCII ROUTINE
MASK4   EQU   3FH        ;MASKS MSBITS OF DATA WORD,USED IN TIME ROUTINE
ZERO3   EQU   374Q       ;MASKS 6 MSBITS OF STATUS WORD,USED IN REVSTA
THREE   EQU   3Q         ;MASKS 2 LSBITS OF STATUS WORD,USED IN REVSTA
MASKC   EQU   60Q        ;MASKS HOUR BIT IN TIME ROUTINE
```

```
;ENTRY VECTORS
        ORG   6000H        ;ABSOLUTE STARTING ADDRESS IN HEX
        DB    50H          ;ALTERNATE I/O CODE PRESENT
        DB    70H          ;CHECK FOR CORRECT LOCATION
        JMP   INIT2        ;INITIALIZATION FROM RESET
        JMP   INIT2        ;INITIALIZATION FROM PROGRAM
        JMP   RETURN       ;INTERRUPT NOT USED,RETURN
        JMP   MONIT        ;MONITOR ROUTINE USED TO DECREMENT TIME
        JMP   INPUT        ;DATA INPUT ROUTINE
        JMP   OUTPUT       ;DATA OUTPUT ROUTINE
        JMP   CONTRL       ;ROUTINE TO CONTROL MAGNETOMETER PROGRAM
        JMP   RETURN       ;STATUS NOT USED,RETURN
        JMP   CHTIMO       ;START ADDRESS OF PROGRAM,CHECK FOR TIMEOUT
RETURN  EQU   $
        RET                ;RETURN TO TERMINAL EXEC WAIT LOOP
;READ THE DATA
INPUT   EQU   $
        LDA   INITD        ;SET IN FF ON MS MAGGIE BYTE I/O BD
        LDA   RSTDVM       ;RESET IN FF ON MS BYTE MAGGIE I/O BD,CAPTURE DATA
        LDA   INSTAD       ;INPUT STATUS BYTE ON MS MAGGIE BYTE I/O BD
        STA   STAT02       ;STORE STATUS BYTE
        LDA   INDVM        ;INPUT DATA BYTE ON MSCHAR MAGGIE I/O BD
        STA   DATA02       ;STORE DATA BYTE
        LDA   INITC        ;SET IN FF ON CLOCK I/O BOARD
        LDA   RSTCLK       ;RESET IN FF ON CLOCK I/O BD,CAPTURE TIME(MIN,SEC)
        LDA   INSTAC       ;INPUT STATUS BYTE ON CLOCK I/O BD
        STA   STAT01       ;STORE STATUS BYTE
        LDA   INCLK        ;INPUT DATA BYTE ON CLOCK I/O BOARD
        STA   DATA01       ;STORE DATA BYTE
        CALL  TIMBIT       ;CALL PROGRAM TO REARRANGE TIME BITS
        LDA   INITM        ;SET IN FF ON LS MAGGIE BYTE I/O BOARD
        LDA   RSTMAG       ;RESET IN FF ON LS MAGGIE BYTE I/O BD;CAPTURE DATA
        LDA   INSTAM       ;INPUT STATUS BYTE ON LS MAGGIE BYTE I/O BD
        STA   STAT03       ;STORE STATUS BYTE
        LDA   INMAG        ;INPUT DATA BYTE ON LS MAGGIE BYTE I/O BD
        STA   DATA03       ;STORE DATA BYTE
GETLIT  EQU   $
        PUSH  B            ;SAVE REGISTER INFO IN STACK
        PUSH  D            ;SAVE REGISTER INFO IN STACK
        LXI   B,STAT04     ;LOAD REG B WITH ADDRESS OF NEXT STATUS BYTE
        LXI   D,DATA04     ;LOAD REG D WITH ADDRESS OF NEXT DATA BYTE
        CALL  AGNLIT       ;GET LATITUDE INFO
        CALL  AGNLIT       ;GET LONGITUDE INFO
        CALL  AGNLIT       ;GET HEADING INFO
        CALL  AGNLIT       ;GET SELECTED INFO
        POP   D            ;RESTORE REGISTER D
        POP   B            ;RESTORE REGISTER B
        CALL  NSEW         ;DETERMINE NORTH,SOUTH,AND EAST,WEST
        RET                ;RETURN TO CONTRL
AGNLIT  EQU   $
        LDA   INITL1       ;SET IN FF ON MS LITTON BYTE I/O BOARD
        LDA   INITL        ;SET IN FF ON LS LITTON BYTE I/O BOARD
        LDA   RSTLT1       ;RESET IN FF ON MS LITTON BOARD,CAPTURE DATA
        LDA   INSTL1       ;INPUT STATUS BYTE FROM MS LITTON I/O BD
        STAX  B            ;STORE STATUS BYTE
        INX   B            ;INCREMENT STATUS ADDRESS
        LDA   INSTAL       ;INPUT STATUS BYTE FROM LS LITTON I/O BD
        STAX  B            ;STORE STATUS BYTE
        INX   B            ;INCREMENT STATUS ADDRESS
        LDA   INLIT1       ;INPUT DATA BYTE FROM MS LITTON I/O BD
        STAX  D            ;SAVE DATA BYTE FROM MS LITTON I/O BD
        INX   D            ;INCREMENT DATA ADDRESS
        LDA   INLIT        ;INPUT DATA BYTE FROM LS LITTON I/O BD
        STAX  D            ;SAVE DATA BYTE FROM LS LITTON I/O BD
```

```
        INX   D          ;INCREMENT DATA ADDRESS
WAITLT  EQU   $
        LDA   CHKFF      ;LOAD REG A WITH FLAG FROM LS LITTON I/O BD
        ANI   INFF       ;CHECK FOR RESET OF FLAG(IN FF)
        SUI   INFF       ;
        JP    WAITLT     ;WAIT FOR RESET OF FLAG
        RET              ;RETURN TO PROGRAM CONTRL
COUNT3  DB   0H          ;TEMPORARY STORAGE
DATA01  DB   0H          ;DATA BYTE:SEC OF TIME FROM CLOCK
DATA02  DB   0H          ;          :MAGGIE MS I/O ??XX
DATA03  DB   0H          ;          :MAGGIE LS I/O ??XX
DATA04  DB   0H          ;          :LATITUDE S?XDEG X?.?MIN(S?X X?.?)
DATA05  DB   0H          ;          :LATITUDE SIGN NORTH/SOUTH
DATA06  DB   0H          ;          :LONGITUDE S?XDEG X?.?MIN(S?X X?.?)
DATA07  DB   0H          ;          :LONGITUDE SIGN
DATA08  DB   0H          ;          :HEADING UNITS AND TENTHS DEG ??X.X
DATA09  DB   0H          ;          :HEADING NOT USED
DATA10  DB   0H          ;          :SELECTED FROM LITTON INTERFACE
DATA11  DB   0H          ;          :SELECTED FROM LITTON INTERFACE
STAT01  DB   0H          ;          :MINUTES FROM CLOCK
STAT02  DB   0H          ;          :MAGGIE-NOT USED
STAT03  DB   0H          ;          :MAGGIE-LS I/O XX??
STAT04  DB   0H          ;          :LATITUDE UNITS AND TENTHS MIN(X.X)
STAT05  DB   0H          ;          :LATITUDE -USED TO SAVE SIGN
STAT06  DB   0H          ;          :LONGITUDE UNITS AND TENTHS MIN(X.X)
STAT07  DB   0H          ;          :LONGITUDE HUNDREDS AND SAVE SIGN
STAT08  DB   0H          ;          :HEADING-HUNDREDS AND TENS XX?.?
STAT09  DB   0H          ;          :HEADING-NOT USED
STAT10  DB   0H          ;          :SELECTED FROM LITTON INTERFACE
STAT11  DB   0H          ;          :SELECTED FROM LITTON INTERFACE
```

```
INIT2   EQU   $
        MVI   A,100       ;MOVE IMMEDIATE 100 TO A
        STA   TIMER       ;STORE 100 TEN MILLISEC IN TIMER(1 SEC)
        MVI   A,2         ;MOVE IMMEDIATE 2 TO REG A
        STA   COUNT4      ;STORE 2 IN COUNT FOR REPEAT TIME
        RET               ;RETURN TO CALLING PROGRAM
COUNT4  DB   0H           ;NUMBER OF REPEAT SECONDS
.ROUTINE TO OUTPUT DATA TO DISPLAY AND CTU
OUTPUT  EQU   $
        MVI  A,6          ;MOVE IMMEDIATE 6 TO A REG
        STA OUTDEV        ;SET UP TO OUTPUT TO DISPLAY AND RT TAPE(110)
        CALL GTIOB0       ;GET A SYSTEM BUFFER
        MVI  M,200Q       ;CLAIM BUFFER WITH BIT
        PUSH H            ;SAVE STATUS POINTER
        DCX  H            ;DECREMENT
        MVI  M,377Q       ;SET UP RECORD TRANSFER(-1)
        DCX  H            ;DECREMENT
        MVI  M,36         ;SET LENGTH OF RECORD TO 36
        XCHG              ;SWAP H&L AND D&E
        CALL GETPTR       ;GET BUFFER ADDRESS
        CALL MOVDAT       ;MOVE DATA INTO BUFFER OBTAINED BY GTIOB0
        POP  D            ;RESTORE STATUS POINTER
        LXI  H,XFRLIM     ;TRANSFER ONE RECORD
        MVI  M,-1
        CALL PUTIO        ;OUTPUT THE RECORD
        XCHG              ;SWAP H&L AND D&E REGISTERS
        MVI  M,0          ;RELEASE BUFFER
        RET               ;RETURN TO CALLING PROGRAM CONTRL
DATAWD  DB   0H           ;TEMPORARY STORAGE
STATUS  DB   0H           ;TEMPORARY STORAGE
;ROUTINE TO MOVE DATA TO BUFFER
MOVDAT  EQU  $
        MVI  A,43         ;MOVE IMMEDIATE 43 TO REGISTER A
        STA  COUNT1       ;STORE IT
        LXI  B,ASBCD1     ;LOAD IMMEDIATE ADDRESS OF FIRST ASCII CHAR
SAVMOR  EQU  $
        LDAX  B           ;LOAD ASCII CHARACTER INTO REGISTER A
        MOV   M,A         ;MOVE CHARACTER TO BUFFER FOR OUPUT
        INX   H           ;INCREMENT BUFFER ADDRESS
        INX   B           ;INCREMENT ASCII DATA ADDRESS
        LDA   COUNT1      ;LOAD COUNT VALUE IN REG A
        DCR   A           ;DECREMENT THE COUNT
        STA   COUNT1      ;STORE THE COUNT
        JP    SAVMOR      ;JUMP ON POSITIVE TO MOVE MORE ASCII CHAR
        STC               ;FINISHED SET CONTROL
        RET               ;RETURN TO CALLING PROGRAM OUTPUT
COUNT1  DB    0H          ;CHARACTER COUNT
```

```
.MONITOR ROUTINE FOR TIMING DATA INPUT
MONIT   EQU    $
        LXI    H,TIMER     ;LOAD IMMEDIATE ADDRESS OF TIMER
        DCR    M           ;DECREMENT TIMER
        RET                ;RETURN TO TERMINAL EXEC
;ROUTINE TO CHECK FOR TIMEOUT
CHTIMO  EQU    $
        LDA    TIMER       ;LOAD REG A WITH VALUE STORED AT LOC TIMER
        ORA    A           ;
        JP     RETURN      ;IF TIMER IS POSITIVE RETURN TO TERMINAL EXEC
        LDA    COUNT4      ;TIMER NEG-LOAD REG A WITH REPEAT TIMER COUNT
        DCR    A           ;DECREMENT COUNT
        STA    COUNT4      ;STORE COUNT
        JM     CONTRL      ;IF REPEAT IS NEG TIME TO EXECUTE MAGGIE PROGRAM
        MVI    A,100       ;THREE SECONDS HAVE NOT OCCURRED RESET 1 SEC TIMER
        STA    TIMER       ;SAVE 1 SEC IN TIMER
        JMP    RETURN      ;RETURN TO TERMINAL EXEC
.CONTROL ROUTINE TO GET AND PROCESS DATA
CONTRL  EQU    $
        CALL   INIT2       ;CALL INITIALIZATION ROUTINE(RESCHEDULES PROGRAM)
        CALL   INPUT       ;GET THE DATA
        CALL   PROCES      ;PROCESS THE DATA
        CALL   OUTPUT      ;OUTPUT THE DATA
        RET                ;RETURN TO CHTIMO
;DATA PROCESSING ROUTINE
PROCES  EQU    $
        LXI    B,STAT01    ;LOAD IMMEDIATE ADDRESS OF FIRST STATUS BYTE
        LXI    H,DATA01    ;LOAD IMMEDIATE ADDRESS OF FIRST DATA BYTE
        LXI    D,ASBCD1    ;LOAD IMMEDIATE ADDRESS OF FIRST CHAR BYTE
        MVI    A,10        ;MOVE IMMEDIATE NUMBER OF WORDS TO PROCESS TO A REG
        STA    COUNT2      ;STORE NUMBER OF WORDS
CVTMOR  EQU    $
        LDAX   B           ;LOAD A WITH STATUS BYTE
        STA    STATUS      ;STORE IT TEMPORARILY
        CALL   REVSTA      ;COMPLEMENT STATUS BITS 0 AND 1(XXXXXXCC)
        LDA    STATUS      ;LDA REG A WITH COMPLEMENTED STATUS BYTE
        STAX   B           ;SAVE IT IN STATUS LOCATION
        STA    CNBYTE      ;SAVE STATUS BYTE IN TEMP LOCATION
        CALL   BCD2AS      ;CONVERT STATUS 2 CHAR BCD BYTE TO 2 ASCII CHAR
        LDA    ASMSB       ;LOAD MOST SIGNIFICANT(MS) ASCII CHAR TO REG A
        STAX   D           ;STORE CHAR IN ASCII BUFFER FOR OUTPUT
        INX    D           ;INCREMENT ASCII STORAGE LOCATION
        LDA    ASLSB       ;LOAD LEAST SIGNIFICANT(LS) ASCII CHAR IN REG A
        STAX   D           ;STORE CHAR IN ASCII BUFFER FOR OUTPUT
        INX    D           ;INCREMENT ASCII BUFFER ADDRESS
        XCHG               ;SWAP REGISTERS H&L AND D&E
        LDAX   D           ;LOAD DATA BYTE IN REG A
        STA    DATAWD      ;STORE TEMPORARILY
        CALL   CMPDAT      ;COMPLEMENT THE DATA BYTE
        LDA    DATAWD      ;LOAD THE DATA BYTE INTO THE A REG
        STAX   D           ;STORE THE DATA BYTE INTO DATA LOCATION
        XCHG               ;SWAP H&L AND D&E
        STA    CNBYTE      ;STORE DATA BYTE TEMPORARILY
        CALL   BCD2AS      ;CONVERT BYTE INTO TWO ASCII CHARACTERS
        LDA    ASMSB       ;LOAD MS ASCII CHAR INTO REG A
        STAX   D           ;STORE CHAR IN ASCII OUTPUT BUFFER
        INX    D           ;INCREMENT ADDRESS OF ASCII OUTPUT BUFFER
        LDA    ASLSB       ;LOAD LS ASCII CHAR INTO REG A
        STAX   D           ;STORE CHAR IN ASCII OUTPUT BUFFER
        INX    D           ;INCREMENT ADDRESS OF STATUS BYTE
        INX    H           ;INCREMENT ADDRESS OF DATA BYTE
        INX    B           ;INCREMENT ADDRESS OF ASCII OUTPUT BUFFER
        LDA    COUNT2      ;COUNT TO REG A
        DCR    A           ;DECREMENT COUNT
```

```
         STA     COUNT2      ;STORE COUNT
         JP      CVTMOR      ;CONVERT MORE BYTES TO ASCII EQUIV IF POSITIVE
         STC                 ;SET CONTROL
         RET                 ;RETURN TO CALLING PROGRAM CONTRL
COUNT2   DB      0H          ;COUNT FOR CONVERTING ALL STATUS AND DATA BYTES TO ASC..
CNBYTE   DB      0H          ;TEMPORARY STORAGE
```

```
;ROUTINE TO REARRANGE STATUS WORD BITS
REVSTA  EQU   $
        LDA   STATUS     ;LOAD STATUS BYTE INTO REG A
        CMA              ;COMPLEMENT THE BYTE
        ANI   THREE      ;MASK OFF BITS 0 AND 1
        STA   REVBIT     ;STORE IT
        PUSH  H          ;SAVE CURRENT ADDRESS H REG
        LXI   H,REVBIT   ;LOAD IMMEDIATE ADDRESS REVBIT
        LDA   STATUS     ;LOAD STATUS BYTE INTO REG A
        ANI   ZERO3      ;MASK OFF BITS 2 THRU 7
        ORA   M          ;COMBINE BITS 0,1 AND 2-7
        STA   STATUS     ;STORE NEW STATUS BYTE
        POP   H          ;RESTORE H REGISTER
        RET              ;RETURN TO CALLING PROGRAM PROCES
REVBIT  DB    0H         ;TEMPORARY STORAGE OF STATUS BYTE
;ROUTINE TO COMPLEMENT DATA WORD
CMPDAT  EQU   $
        LDA   DATAWD     ;LOAD A WITH DATA BYTE
        CMA              ;COMPLEMENT DATA BYTE
        STA   DATAWD     ;STORE IT
        RET              ;RETURN TO CALLING PROGRAM PROCES
;BCD TO ASCII CONVERSION ROUTINE
BCD2AS  EQU   $
        LDA   CNBYTE     ;LOAD REG A WITH DATA BYTE IN BCD FMT
        RRC              ;ROTATE RIGHT FOUR TIMES
        RRC              ;
        RRC              ;
        RRC              ;
        ANI   MASK1      ;MASK OFF BCD CHAR
        ACI   30H        ;ADD 30 HEX TO CHAR TO CONVERT TO ASCII
        STA   ASMSB      ;STORE MOST SIGNIFICANT ASCII CHAR
        LDA   CNBYTE     ;LOAD A WITH DATA BYTE IN BCD FMT
        ANI   MASK1      ;MASK OFF BCD CHARACTER
        ACI   30H        ;ADD 30 HEX TO CHAR TO CONVERT TO ASCII
        STA   ASLSB      ;STORE LEAST SIGNIFICANT ASCII CHAR
        RET              ;RETURN TO CALLING PROGRAM PROCES
ASMSB   DB    0          ;TEMP STORAGE
ASLSB   DB    0          ;TEMP STORAGE
;ROUTINE TO SET UP BITS FOR NS AND EW
NSEW    EQU   $
        LDA   STAT04     ;LOAD REG A WITH LATITUDE STATUS BYTE
        RRC              ;ROTATE RT 2 BITS
        RRC              ;THESE ARE THE SIGN BITS
        ANI   MASKC      ;MASK OFF THESE SIGN BITS
        CMA              ;COMPLEMENT THE SIGN BITS
        STA   DATA05     ;SAVE THE BITS IN THE LAST BYTE OF THE LAT WORD
        LDA   STAT04     ;LOAD REG A WITH LATITUDE STATUS BYTE
        ANI   MASK4      ;MASK OFF DEGREES LATITUDE WITHOUT SIGN
        STA   STAT04     ;STORE DEG LAT IN STATUS BYTE
        LDA   STAT06     ;LOAD REG A WITH LONGITUDE BYTE
        RRC              ;ROTATE RT 2 BITS
        RRC              ;GET SIGN BITS
        ANI   MASKC      ;MASK OFF SIGN BITS
        CMA              ;COMPLEMENT SIGN BITS
        STA   DATA07     ;STORE SIGN BITS IN LAST 2 CHAR OF LONGITUDE WORD
        LDA   STAT06     ;LOAD REG A WITH LONG BYTE
        ANI   MASK4      ;MASK OFF DEG LONG WITHOUT SIGN
        STA   STAT06     ;STORE DEG LONG IN STATUS BYTE
        LDA   STAT08     ;LOAD REG A WITH HEADING BYTE
        RRC              ;ROTATE RT 2 BITS
        RRC              ;GET SIGN BITS
        ANI   MASKC      ;MASK OFF SIGN BITS
        CMA              ;COMPLEMENT THE SIGN BITS
        STA   DATA09     ;STORE SIGN BITS IN LAST BYTE OF HEADING WORD
```

```
         LDA    STAT08     ;LOAD REG A WITH HEADING BYTE
         ANI    MASK4      ;MASK OFF THE BYTE WITHOUT SIGN
         STA    STAT08     ;SAVE HEADING IN STATUS BYTE
         RET               ;RETURN TO CALLING PROGRAM PROCES
;TERMINAL AIRMAGNETICS SYSTEM PART 2,V101 12/04/79
;THIS ROUTINE REARRANGES TIME BITS
MIN1        EQU    77Q     ;MASK FOR MINUTES OF TIME
SEC1        EQU    177Q    ;MASK FOR SECONDS OF TIME
SEC2        EQU    200Q    ;MASK FOR MINUTES OF TIME SECOND WORD
TIMBIT      EQU    $
            LDA    DATA01  ;LOAD REG A WITH SECONDS OF TIME
            CMA            ;COMPLEMENT DATA
            STA    DATA01  ;STORE IT
            LDA    DATA01  ;
            ANI    SEC1    ;MASK OFF SECONDS OF TIME
            STA    DATAC1  ;STORE IT
            LDA    DATA01  ;LOAD REG A WITH SEC DATA BYTE
            ORA    A       ;
            ANI    SEC2    ;MASK OFF BIT ASSOCIATED WITH MINUTES(MSBIT)
            ORA    A       ;
            RLC            ;ROTATE BIT LEFT ONCE,PUTS IT AT BIT ZERO
            STA    DATAC2  ;STORE MINUTES BIT
            LDA    DATAC1  ;LOAD REG A WITH SEC BYTE
            CMA            ;COMPLEMENT BYTE
            STA    DATA01  ;RETURN BYTE TO DATA WORD
            LDA    STAT01  ;LOAD A WITH MINUTES BYTE
            STA    STATUS  ;STORE TEMP
            CALL   REVSTA  ;REVERSE BITS 0 AND 1
            LDA    STATUS  ;LOAD REG A WITH MINUTES BYTE
            STA    STAT01  ;STORE MINUTES IN STATUS BYTE
            LDA    STAT01  ;
            ORA    A       ;
            ANI    MIN1    ;MASK OFF MIN FROM STATUS BYTE
            ORA    A       ;
            RLC            ;ROTATE LEFT ONCE;GET READY TO FORM NEW MIN WORD
            ORA    A       ;
            STA    STAT01  ;SAVE MIN SHIFTED
            LXI    H,DATAC2;LOAD IMMEDIATE ADDRESS OF MINUTES BIT
            LDA    STAT01  ;LOAD REGISTER A WITH MIN BYTE
            ADD    M       ;ADD MINUTES BIT TO MIN BYTE
            STA    STAT01  ;STORE FULL MINUTES WORD IN STATUS BYTE
            LDA    STAT01  ;
            STA    STATUS  ;STORE TEMP
            CALL   REVSTA  ;PLACE MINUTES BYTE IN USUAL FORMAT
            LDA    STATUS  ;LOAD MINUTES BYTE INTO REG A
            STA    STAT01  ;STORE MINUTES BYTE INTO STATUS BYTE IN RECONSTR FMT
            RET            ;RETURN TO CALLING PROGRAM INPUT
DATAC1      DB     0H      ;TEMPORARY LOCATION
DATAC2      DB     0H      ;TEMPORARY LOCATION
TEMP10      DB     0H      ;TEMPORARY LOCATION-NOT USED
;SETUP      OUTPUT WORDS
ASBCD1      DS     18      ;ASCII FILE SET UP FOR OUTPUT
ASBC19      DS     8       ;
ASBC27      DS     8       ;
ASBC35      DS     8       ;
ASBC43      DS     8       ;
            END
```

APPENDIX II

OPERATING INSTRUCTIONS FOR THE TERMINAL MAGNETICS FIELD DATA ACQUISITION SYSTEM

1. Turn on power to the terminal, Litton Interface and Clock.

2. Set the thumbwheel switch on the Litton Interface to "0".

   Explanation: By setting the select code on "0" Latitude will be selected and displayed on the interface. The terminal receives latitude information from the display.

3. Insert cartridge tape marked Debugger/Assembler in left tape drive of terminal.

4. Press the key marked READ on the terminal. Wait for completion.

   Explanation: The first record of the Debugger/Assembler tape will be displayed.

5. Press the key marked f2 on the terminal. Wait for completion.

   Explanation: By pressing f2 the second record on the Debugger/Assembler tape will be loaded into the terminal memory. The message "OK>" will be displayed on the terminal.

6. Remove the Debugger/Assembler tape from left drive and insert the tape marked Magnetics Version 13 Binary.

   Explanation: This is the binary magnetics program to be loaded into terminal memory.

7. Type the characters "L" and "CR" (Carriage Return). Wait for completion.

   Explanation: This sequence will load the binary program into terminal memory. The message "HP264X ASSEMBLER V2.0" will appear on the terminal display followed by an "OK>" prompt.

8. Place a blank cartridge in the right terminal drive.

Explanation: The data will be recorded on this tape cartridge. The cartridge should be unprotected by moving the protect lever to the left position. The tape cartridge should be labeled by hand. The recommended labeling is day of year and starting hour of tape.

9. Type "/9169" then "CR" on the terminal.

Explanation: An instruction in location $9169_{(16)}$ must be modified so that control will be transferred from the terminal executive software to the magnetics program. An "87" will appear on the display.

10. Type "601A" then "CR" on the terminal.

Explanation: The starting location of the magnetics program is $601A_{(16)}$. An "0" will appear on the display.

11. Type ":" (colon) on the terminal.

Explanation: The ":" will terminate the modification process. An "OK>" will appear on the terminal display.

12. Press the RESET button on the terminal only once.

Explanation: Pressing the RESET button once forces a transfer in the terminal executive to the magnetics program. The program will start execution.

REFERENCES

1. HP13255 Terminal Duplex Register Module Manual, Hewlett Packard Part Number 13255-91031.

2. HP13290B Debugger/Assembler Reference Manual, Hewlett Packard Part Number 13290-90009.

3. Steiger, D., "Using Intelligent Graphics Terminals in Real-Time Processing", NRL Memorandum 4055 (August 24, 1979).

4. Clamons, J. D. and Steiger, D., "Can Intelligent Terminals and Modern Calculators Replace Oceanographic Computer Systems?", Woods Hole Oceanographic Institution Proceedings, Second Working Conference on Oceanographic Data Systems (September 1978).